AUGMENTATION OF IoT FRAMEWORK IN BUILDING MANAGEMENT SYSTEM

SHORT SYNOPSIS

Submitted in partial fulfillment of the requirement of the degree of

DOCTOR OF PHILOSOPHY

to

Manav Rachna International Institute of Research and Studies
(Deemed to be University)

by

Uma Tomer
19/Ph.D/025

Under the supervision of
Dr. Parul Gandhi
Professor, Faculty of Computer Applications

FACULTY OF COMPUTER APPLICATIONS

MANAV RACHNA INTERNATIONAL INSTITUTE OF RESEARCH & STUDIES

SECTOR-43, SURAJKUND –DELHI ROAD, FARIDABAD – 121001

FACULTY OF COMPUTER APPLICATION

Certificate

I hereby declare that the short synopsis "Augmentation of IoT Framework in Building Management System" being submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy in Computer Applications under faculty of Computer Applications of Manav Rachna International Institute of Research & Studies, Faridabad during the academic year 2021-2022.

It is a bona fide record of original work carried out under the supervision of Dr. Parul Gandhi and has not been submitted in full or part to any other university or institute for degree or diploma. Further, it is certified that short synopsis is plagiarism free and proof has been submitted.

(Uma Tomer)
19/Ph.D/025

Dr. Parul Gandhi
Professor,
Faculty of Computer Applications

(Chairman, DRC)

(Director, CDP)

ABSTRACT

Internet of Things (IoT) is growing up promisingly in this world due to its wide range of applications. IoT consists of small, tiny nodes which comprise of microcontroller units, sensors and actuators, and any communication device (Wi-Fi, Bluetooth, Zigbee, etc.) which helps to connect to the internet. Here, we have explored the existing IoT frameworks and architectures utilized in Building Management System (BMS) and home automations. Wide range of commercial hardware, Internet connectivity, different network topologies, and large application space forces the research community to work on the Quality of Services (QoS) for BMS applications. In last decades, it is reviewed that IoT has been implemented in large scale for Building Management Systems. Optimization of energy and power due to improvement in QoS in IoT, is the motivation of research for BMS applications. Enhanced framework for QoS in IoT is still a challenge for the research community because of diversity in IoT frameworks for different applications. An enhanced framework of IoT for improved QoS will be proposed and implemented in the future research work. Contiki and Mongoose operating system (OS) are well known OS for IoT framework and may be considered in the research to evaluate the parameters of QoS of proposed IoT framework. In methodology, first, the close literature review will be conducted for understanding and defining the parameters for QoS, then an analysis of existing techniques will be done using the software and hardware tools to propose an enhanced framework for IoT in BMS applications. The proposed framework will be verified and implemented in the real time BMS application.

+

Key words: Internet of Things (IoT), Quality of Service (QoS), Network, Topology, Sensors, Actuators, Building Management System (BMS), WiFi, Bluetooth, Zigbee

CONTENTS

S. No.	Description	Page No.
1	Introduction	1-2
2	Literature Review	2-8
3	Description of Broad Area	8-13
4	Objectives of the Study	13
5	Research Methodology	13-14
6	Expected Outcome of the Research	14
	References	
	Plagiarism Report	

1 INTRODUCTION

The Internet of Things (IoT) is one of the most rapidly advancing technologies that have emerged in the past decade, with an increase of 12 percent in the number of businesses that uses IoT technologies.

The main components of the IoT are: Things (object), Communication and Computing.

- **"Things"** can be perceived as real-life objects containing sensors (attached or embedded) that enable these to interact with some external environment.

- "**Communication"** is the central networking element that allows these objects or "things" to interact with one another and/or some external environment with the help of communication protocols based on some network, such as a wired connection, or wireless protocols like WAN, LAN, or cellular.

- "**Computing"** directs the behavior and is handled on the mobile devices, desktop workstations, or the server, which processes and reviews the mined data. Computing makes informed decisions and inferences within a specific system that eventually forms the connection for actual use and user behavior analysis.

In Building Management System (BMS), IoT can be used in every aspect to barbarize its functionalities. The additional benefit of using IoT is its low deployment cost as well as low maintenance cost in the Building Management System at the application level. Demand Response (DR) arrangements are implemented by BMS as it controls and manages different mechanical/electrical systems and also communicates with hardware in all mechanical or electrical systems to monitor the energy consumption in real-time (Javed, Afzal, Sharif and Kim, 2018). Javed, Afzal, Sharif and Kim (2018) mentioned that on global basis a typical building which may be residential or commercial exploits approximately 40% of the total consumption of energy. They also highlighted that on an average electricity consumed per square foot is 15.9 kilowatt-hours (KWh) annually which is equal to the expenditure of about $1.7 at the rate of $0.1058/KWh this shows that the net increment in the rent is about 4-5%.

Iwendi, Maddikunta, Gadekallu, Lakshmanna, Bashir and Piran *(2020)* reviewed different IoT based operating systems on the basis of different key parameters and finally concluded that Contiki

is found to be the most appropriate operating system for maximum Internet of Thing's applications as it uses Low Power Wireless Personal Area Network

(6loWPAN) protocols for communicating which gives real-time functionalities. Other than Contiki

Operating System, Mongoose Operating System have been examined for evaluation in the application of Building Management System as Mongoose Operating System comes with various features which includes easy communication with cloud, support of modern programming language, low latency, secure connectivity, crypto support, integrated mongoose web server, and multithreading support.

Though Contiki was released in 2004, it is still in well interest of researchers because it has some valuable features that includes low RAM, low ROM, C programming support, multithreading (optionally), Microcontroller Unit (MCU) supported, modular approach, and suitable for real-time application. Contiki Operating System comes with a number of versions and is Berkeley Software Distribution (BSD) licensed. Contiki uses the Rime networking approach of version ak. Contiki also supports Low Power Wireless Personal Area Networks (6LoWPAN) (Iwendi, Maddikunta, Gadekallu, Lakshmanna, Bashir and Piran , *2020)*.

Mongoose Operating System is an IoT framework which uses *vendor's* Software Development Kit (SDK), that extends the limitation of native SDK. In microcontrollers like ESP32, Mongoose Operating System utilizes *ESP-idf* SDK which enables all the features *ESP-idf* SDK and also gives some extra features that comes under Mongoose Operating System itself.

The discussed operating systems helps to enhance the framework and is an effective tool in enhancing the QoS of IoT systems, embedded systems etc. They are based on Real Time Operating System (OS) framework and also provide good opportunities to adapt different technology and applications of IoT.

2 LITERATURE REVIEW

Abdallah, Jaber, Alabwaini and Alnabi (2019) have studied various quality models for IoT systems and compared them with several quality factors. Besides this, a new model has been introduced, focusing on IoT's characteristics by introducing quality factors that measure them. Furthermore, the

researchers added that the proposed model would improve quality factor which will affect IoT system quality. In future scope, they added a requirement of dynamic quality model which would give more weight on each IoT characteristic, which would depend on the nature of IoT.

Aldrich, Chambers, Sirer and Eggers (1999) have presented a set of reports to decrease synchronization overhead in Java programs due to monitor-based synchronizations. To reduce these unnecessary lock operations, manual optimizations have been replaced by static analyses that automatically detect and eliminate non-essential synchronization from Java programs.

Ammar, Russello and Crispo (2018) have done a survey on security of the IoT frameworks. The proposed architecture of each framework, compatible hardware and security features have been discussed.

Asghari, Rahmani and Javadi (2020) proposed a cloud service approach that is privacy aware, corresponding to optimization via QoS in IoT systems through a conceptual model of privacy level and a unique hybrid evolutionary algorithm using the Shuffled Frog Leaping Algorithm (SFLA) and Genetic Algorithm (GA) called the SFLA-GA. This algorithm optimizes the service composition by aggregating different QoS factors as one fitness value.

Bogda and Holzie (1999) have described an intraprocedural flow and context insensitive dataflow analysis that finds unnecessary synchronizations in Java programs. The framework optimizes more than 90% of the alias sets containing a minimum of one synchronized object for each program in their suite of 10 Java benchmarks, containing SPECjvm98 and others.

Baccelli, Hahm, Gunes, Wahlisch and Schmidt (2013) compared different existing operating systems for powering IoT devices and concluded that none performs well in all the parameters and pointed out some challenges faced during implementing different IoT OS, some the challenges are Operating System should be able to make connects using various protocols like Hypertext Transfer Protocols (HTTPS), Message Queuing Telemetry Transport (MQTT), Machine-to-Machine (M2M), etc. with cloud support. Memory constraint, as most of the IoT devices have very less memory which is barely 5 kB that's why the minimum requirement of the memory by the OS should be very low. Further they introduced an IoT operating system named RIOT, which performs well on the parameters and fills all the requirements. It is based on a microkernel architecture which is found reliable. Some other features of RIOT includes that its kernel requires less Random-Access Memory (RAM) and Read-Only Memory (ROM), developer friendly interface, supports large number of platforms and

works well with existing libraries. RIOT provides a reliable microkernel architecture, where the kernel itself requires just a few hundred bytes of ROM and RAM. Among its features it comprises a highly adaptive network stack, including the latest standards of the Internet Engineering Task Force (IETF) for connecting constrained systems to the Internet like 6LoWPAN and Routing Protocol for Low-Power and Lossy Network (RPL). Due to RIOT's developer-friendly Application Programming Interface (API), which is partly Portable Operating System Interface (POSIX) compliant, and support for various platforms, it is possible to build the whole software system upon RIOT OS and easily adopt existing libraries for projects involving introducing a novel operating system, RIOT, which meets these requirements.

Cuka, Elmazi, Obukata, Ozera, Oda and Barolli (2017) have made use of Fuzzy Logic and Genetic Algorithm to design an integrated intelligent system for IoT device selection. They have introduced the structure of the system and presented the design and implementation issues in detail.

Cheruvu, Kumar, Smith and Wheeler (2019) surveyed various IoT Frameworks which have been classified based on their application into consumer, industrial, or manageability. The have focused on the complex architecture that lies underneath as the industry focuses on newer Internet protocols keeping in hand the backward compatibility.

Djaker, Kechar, Ibn-Khedher, Moungla and Afifi (2020) have proposed a new algorithm - Distributed Maximum Concurrent Flow (DMCF) to overcome the problems of Data streaming over IoT network. The proposed algorithm urges two different approaches, one using graph theories and other based on Reinforcement Learning techniques. Performance has been estimated over many complex graphs, and the conclusion is that reinforcement learning performs better than conventional graph theories.

Goudarzi, Wu, Palaniswami and Buyya (2020) stated that due to the increasing growth of IoT devices and fog/edge servers with restricted resources, these cannot fulfill the requirements of IoT applications. The challenge was to place fog/edge computing environments and centralized cloud servers. To tackle this, they have suggested a cost weighted model that decreases the energy consumption and execution time of IoT applications. In addition to that, a Memetic algorithm that decides batch application placements for concurrent programs has been proposed. To maximize the concurrency and handle the homogeneity of IoT applications, a light-weight pre- scheduling algorithm has been proposed. Results have shown an improvement of upto 65% as compared to its counterparts.

Hassan and Noor (2019) presented an examination of recent IoT security research from 2016 to 2018, as well as patterns and open problems. Their paper provides a summary of research in IoT security, its relevant tools and IoT simulators and modelers.

Iwendi, Maddikunta, Gadekallu, Lakshmanna, Bashir and Piran (2020) have focused on minimizing the energy consumption by sensors in IoT networks which leads to an increase in the network lifetime. Focusing on Cluster Head (CH), they have proposed a hybrid metaheuristic algorithm Whale Optimization Algorithm (WOA) with Simulated Annealing (SA). Several metrics have been considered like the total alive nodes, temperature, load, cost function and residual energy. This algorithm is then compared with existing algorithms such as Artificial Bee Colony Algorithm, Genetic Algorithm, WOA and Adaptive Gravitational Search Algorithm.

Izci, Ekinci, Eker and Kayri (2020) have made use of Manta Ray Foraging (MRF) along with Opposition based Learning (OBL) to come up with a good foundation for solving optimization problems. Opposition-based Manta ray foraging optimization is the name assigned to the proposed structure. The introduced algorithm was tested on four benchmark algorithms and compared with other optimization algorithms.

Javed, Afzal, Sharif and Kim (2018) researched that the primary requirements for different applications of IoT are architectural, projected algorithms, language provision, power handling and memory requirement, concluding that on the basis of different key parameters and finally concluded that Contiki and RIOT are most widely implemented IoT operating system because of their numerous features. According to them an operating system should provide instantaneous features and 6loWPAN protocols for the purpose of communication. Further they concluded that privacy, security and interoperability are the most commonly faced challenges in IoT.

Kawachiya, Koseki and Onodera (2002) have suggested the Lock reservation algorithm. It makes use of Java's thread locality, which states that Java's locking sequence is a very long and repeated process. When a thread tries to obtain a lock, if the lock is reserved for the thread, it will do so without doing any atomic operations. In actual Java programs, they were able to increase performance by up to 53%.

Kiruthika and Khaddaj (2015) have addressed the various Quality of Service (QoS) issues and the important factors that must be considered while designing the quality models since non-functional factors play crucial role in examining various aspects of a system.

Li (2020) has studied the Building Management System of a company in Singapore and proposed as Economical Building Management System. In this system he has used building management system and machine learning. He proposed an economical decision-making system for Building Management System which take decisions in the absence of a concerned person and monitor real time energy prices and forecast prices of energy. His solution reduces the cost up to 61.42%.

Lu and Cecil (2016) contour a relational structure that serves as a foundation for advanced manufacturing and cyber-physical interactions. To design and model IoT based collaborative framework, Extended Enterprise Modeling Language (EEML) is used. The various cyber physical components and modules are also been discussed for the implementation of the framework.

Minoli, Sohraby and Occhiograsso (2017) mainly described Building Management Systems (BMS) and how to apply BMS in the network. They investigated several emerging technologies in BMS applications which are Power over Ethernet (PoE), IoT, cloud services, and converged networks which will in turn help to get a better perspective on energy conservation in BMS that would finally save expenses on the other hand would help to preserve environment. Further in this paper they looked at the mongoose operating system and how one can make the network more reliable and the system more dynamic. They concluded that the growth of cloud-based device analytics of good quality will enable global optimization and suitable for data mining, sheering and foretelling at a greater level.

Oti, Kurul, Cheung, Tah (2016) reviewed Building Information Modelling (BIM). By mathematical prediction the author says BIM helps in reducing error, project delivering time and save implementation cost. BIM helps in improving design and facility management.

Patel and Patel (2016) have discussed briefly about IoT, How IoT enables other technologies, its architecture, characteristics and applications along with the future challenges.

Ren, Zhang, Chen and Abnoosian (2020) proposed a new algorithm which consolidates Simulated Annealing (SA) and Particle Swarm Optimization (PSO) algorithms in order to overcome the problems of Task Allocation. The distinctive issues of both the algorithms like unable to achieve optimal response and getting stranded in local optimization compelled them to propose this algorithm that overcomes the individual problems of these algorithms. Simulation results of PSO-SA algorithm have shown better performance than the PSO and SA based models.

Sangaiah, Hosseinabadi, Shareh, Rad, Zolfagharian, and Chilamkurti (2020) have used Whale Optimization Algorithm (WOA) to resolve resource allocation between the resources and gateways. The results have shown better results in terms of "total communication cost".

Singh and Baranwal (2018) discussed the QoS metrics of IoT with Computing, Communication and Things as the three pillars of IoT. Keeping in mind the importance of QoS metrics, this work helps professionals, researchers to develop models in IoT based on their needs.

Sivaraman, Gharakheili, Vishwanath, Boreli and Mehani (2015) have illustrated the increasing uptake of smart home appliances and have discussed the challenges that occur due to the emergence of more such devices. They have further proposed a software defined technology that can use these devices based on their network activity.

Suryanegara, Prasetyo, Andriyanto and Hayati (2019) proposed a framework that could interconnect and support various services and public facilities like smart cities. Their proposed framework included five steps. To test their proposed framework, they measured the Quality of Experience (QoE) for IoT services in Jakarta Smart city.

Tian, Luo, Qiu and Guizani (2020) designed a web attack detection system by analyzing URLs which is deployed on edge devices. Zd g. Many parallel deep models have been used to improve the system's reliability. They have achieved an accuracy of 99.410%, 98.91% in TPR and 99.55% in DRN.

Weber (2009) discussed the issues of governing regulatory frameworks, their governing principles, and the implementation of an independently managed decentralized multiple-root system.

White, Nallur and Clarke (2017) reflects on the present scenario of proposed QoS methods in IoT areas such as (1) Which levels of the IoT architecture have had the most QoS research? (2) When assessing results, what quality variables do the quality methods consider? (3) What kinds of studies have been done in this field?

Wijayasekara, Linda, Manic and Rieger (2014) has studied abnormalities in Building Energy Management System [BEMS] and uncertainties in the behavior of the indoor environment. Using the fuzzy logic clustering algorithm and rule extraction technique, they proposed a fuzzy Anomaly Detection and Linguistic Definition (fuzzy-ADLD) to better understand BEMS and improve abnormality detection. They added implementation of the developed software instance on a portable

device such as smart phones in the future scope of the paper, which would establish a portable touch-screen skillful tool. Such organization can aid in the provision of more thorough explanations of the building's behavior as well as potential solutions.

Viriyasitavat, Xu, Bi, Hoonsopon and Charoenruk (2019) introduced a secure Blockchain method for the measurement of various QoS parameters as the traditional methods or mechanism for QoS measurement but lacks to provide trust in monitoring, collecting, accessing the QoS information. Further they developed a full proof secure system to measure the quality-of-service parameters using blockchain models. They are able to promise trust to the end user in the measurement of quality-of-service parameters.

Zhao, Zhang and Wang (2020) have presented a new algorithm which makes use of the intelligent behavior of the Manta Rays. The performance results of this algorithm have far better results than the existing optimizers. This algorithm has proved to be more competitive in solving real-world problems with better cost computations and precision. They have also compared their results with the extant optimizers on real-world problems.

3 DESCRIPTION OF BROAD AREA

3.1 ARCHITECTURE OF IoT

IoT architecture can be understood as an assortment of numerous elements which include sensors, cloud services, actuators, layers etc. these elements when combined make up the complex architecture of IoT as shown in Fig.1. Understanding the architecture of IoT is important to establish stable interconnectivity between the various components and providing best user experience.

The basic layers that make up the IoT model architecture are:

- IoT system device layer
- IoT communication Gateway layer
- IoT based Platform Layer

These three basic layers of IoT can be further divided into four architectural layers. Without considering the needs and requisites, the architecture of IoT is a failure. These layers are:

- Controllers & Sensors
- Data Collection System & Internal Gateways
- Edge Information Technology
- Cloud & data Collection centers

i. Controllers & Sensors – The distinguishable property of sensors to interpret the information obtained from the outside world into data helps to analyze data for further processing of the information. Actuators further help to interpret the physical capacity to gain necessary insights

ii. Data Collection Systems & Internal Gateways – IoT devices are so named because they must be connected to the Internet in order to function and process data. This layer connects the sensors and actuators to the Internet to collect the information and reduce its size to optimal size for further inspection. This stage makes the data both digitalized and aggregated.

iii. Edge Information Technology – At this stage, the data is prepared to be sent to the IT world to perform enhanced analytics. These IT systems are located in close proximity to the sensors and actuators in order to create a wiring closet.

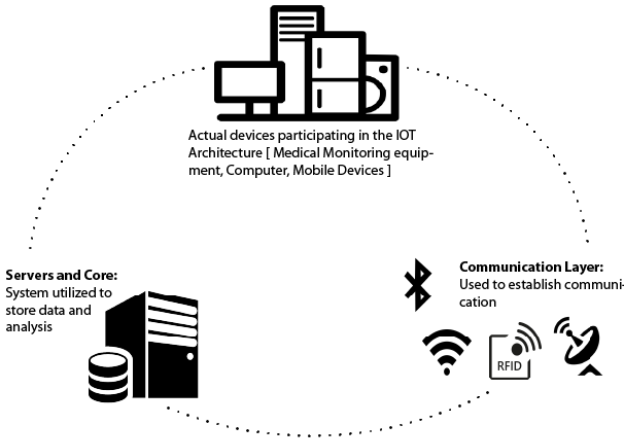
Actual devices participating in the IOT Architecture [Medical Monitoring equipment, Computer, Mobile Devices]
Servers and Core: System utilized to store data and analysis
Communication Layer: Used to establish communication
RFID

Figure 1. Basic Architecture of IoT (Source: Self)

iv. Cloud & data Collection centers – In this stage both the Information Technology (IT) and Operational Technology (OT) work together to perform in-depth processing of the data which

is accompanied by a follow-up revision for feedback. The information processed here meets the quality standards and requirements that can now be transferred to the physical world.

3.2 OPERATING SYSTEMS IN IoT APPLICATIONS

Operating System plays very important role in defining the QoS for IoT services and research. CONTIKI and MONGOOSE are two OS which are popular in research community. CONTIKI already has been adopted by the research community but the MONGOOSE still has scope of research for IoT and may help to find new paradigm for QoS of IoT software and services.

3.2.1 CONTIKI

Modular architecture of IoT is followed by Contiki Operating System as it consists of a mixture of two different models which are event and thread models and another reason for using this approach is its kernel which is small in size that supports different embedded operating systems and context switching of different wireless sensor network applications. The reason for its small sized kernel is that this modular approach only provides minimum or less function to be performed by the kernel. The Advantage of using this operating system is that the functionalities are provided by user-level servers; these functionalities include file, memory, time server etc. which is why this operating system does not easily crash if any single server fails to perform. Contiki as operating systems can provide multi-threading optionally to some applications like application-level libraries as they may require, however Contiki is basically a single threaded operating system at the basic kernel level. Its kernel sends various events to running processes and sometimes refers to processors as polling controllers; all of this is made possible by a lightweight event scheduler.

3.2.2 MONGOOSE

Mongoose Operating System is basically a framework that provides the capabilities to build different applications to be deployed on low-power devices like microcontrollers and its main components are mentioned below.

- A tool that provides device management and capabilities like firmware building etc. and the tool is named as mos tool.
- It is a combination of hardware vendor Software Development Kit (SDK) and Mongoose-Operating System source which is embedded in a docker image named a build toolchain.

- A vast number of libraries and built-in apps.

Mongoose Operating System is an IoT framework which uses *vendor's* SDK, that extends the limitation of native SDK. In microcontrollers like ESP32, Mongoose Operating System utilizes *ESP-idf* SDK which enables all the features *ESP-idf* SDK and also gives some extra features that comes under Mongoose Operating System itself.

Mongoose Operating System provides a complete environment for prototyping, development and managing connected devices which reduces the time and cost associated with IoT projects. It has some of the important features which are mentioned below.

- It is user-friendly Over the Air (OTA) with updating of embedded Integrated Circuits (IC) which provides secure connectivity and crypto support with integrated Mongoose Web Server.
- As it has a networking library it implements event-driven non-blocking APIs for TCP, UDP, HTTP, WebSocket, MQTT.
- Programming of these software's is done using JavaScript or C.

It has a wide range of commercial applications due to its features and vast number of libraries and built-in apps. It is used to solve a wide range of business needs and problems like implementing Web UI interface, remote control for a product, remote software updates, remote monitoring, RESTful API service and telemetry data exchange.

For security purposes, it has built in flash encryption, crypto chip support, Arm Mbed TLS optimized for small memory footprint. As it has Over-The-air firmware it had reliable updates for rollback on failures and remote device access infrastructure. It also consists of a Built-in-integration for AWS IoT, Google IoT Core, Microsoft Azure, Adafruit IO, generic MQTT servers. There is Mongoose respiratory which is integrated into Google's oss-fuzz continuous buzzer which scans for potential vulnerabilities continuously. It also provides specific security requirements to run independent security audits.

For commercial use, it is used by several businesses all across the globe like Siemens, Bosch, Google, Samsung, Qualcomm, Caterpillar to the small businesses. It is developed to hundreds of millions of devices in production environments worldwide.

3.3 FUNCTIONALITY OF QUALITY ASSURANCE WITH IoT

IoT is a network of physical objects whose components like sensors and actuators interact with each other and the existing environment to take knowledgeable results. There are three different components that make up the Core of IoT which are:

- Things – This includes the components like sensors, actuators and installed software that are required for taking intelligent decisions with the extrinsic environment.
- Communication – Communication plays an important role in IoT for best-in-class user experience and smooth functioning of the device. The protocols of communication depend on the type of network like WAN, LAN or PAN. The communication can be wired as well if possible.
- Computing – Computing can be done on various devices depending on the amount of data that is required to be processed for analyzing. It is usually done on mobile phone, desktop or server. This takes place in two stages: the first step takes brainy decisions within the system while the other step provides important links for Big Data Analysis.

Bogda and Holzie (1999) has introduced some of the major challenges faced in IoT and proposed power over the ethernet for better connectivity and solutions to enhance the security and Quality of Services for better utilization of the devices. Minoli, Sohraby and Occhiogrosso (2012) explains the significance of IoT and its application in our lives as well as the industrial sector. IoT has now penetrated every area of human need, including but not limited to smart homes, buildings, and cities. They have been reached to e-health, physical security, street light controlling, meter readings for electricity and gas providers and surveillance etc. A cost-effective IoT solution is needed for healthy growth of IoT and better security for its protection. We are enhancing the possibilities of IoT and wiz-a -wiz we are also needed to enhance the security of the IoT devices.

(Singh and Baranwal *,2018;* Sangaiah, Hosseinabadi, Shareh, Rad, Zolfagharian and Chilamkurti, *2020)* explained different types of monitoring system, management system and optimization of different resources required in commercial building. They required span energy management, access management, video surveillance and monitoring of the environment which also provided fire detection for the commercial building. Smart Buildings are extended versions of smart homes with higher scalability and higher security for devices with easy communication networks within the devices.

Aldrich, Chambers, Sirer, Eggers (1999) has explained the heterogeneous structure of the network and explains the difficulties in network structure for IoT. They proposed a 3-layer scheduling model of Quality of Service (QoS) for service-oriented IoT. The application layer scheduling scheme finds the ideal composition of each service for better QoS-aware service. In the network layer it handles the environment of heterogeneous networks and in the sensing layer it gathers information and allocates it according to the resources available and services. This architecture utilizes the scheduling of services for an ideal Quality of Service (QoS) system and reduces the cost of services.

Zhang, Shao, Ji, Qiu, Tian, Du and Guizani (2020) pointed out some challenges faced during implementing different IoT OS, some of these challenges are listed below:

- Memory constraint, as most of the IoT devices have very less memory which is barely 5KB that's why the minimum requirement of the memory by the OS should be very low.
- Since the clock cycle of most of the MCUs in IoT devices is very low so the complexity of various operations must be kept very low.
- Operating Systems should be able to make connections using various protocols like HTTPS, MQTT, M2M, etc. with cloud support.
- The Operating system must be able to optimize the power of the IoT device and must have a large sleep cycle.

4 OBJECTIVES OF THE STUDY

4.1 PROBLEM STATEMENT

"Distributed architecture of the IoT forces us to work on the need of an enhanced framework which may provide better quality. Quality Control in Building Management Application is still an unattended area in IoT. There are many issues like real time handling, throughput and latency which still need to be improved to meet the quality for BMS."

4.2 RESEARCH OBJECTIVES

- To explore and investigate the optimization algorithms for quality enhancement of IoT frameworks.
- To propose an enhanced IoT framework for Building Management System.

- To evaluate and validate the performance of proposed framework on various key Parameters like latency, memory management etc.

5 RESEARCH METHODOLOGY

The proposed research will include investigation of existing quality model and optimization algorithms for quality enhancement of IoT devices and their associated software. The subsequent steps will include implementation of existing techniques for quality measurement for the IoT Building Management applications, which is based on parameters like latency, memory management, security, real-time through-put and more. Further steps include to propose an enhanced IoT framework with the use of software to design and to develop innovative techniques for quality improvement in the software-based framework for Building Management application. Finally results will be analyzed and the performance evaluation of the framework will be done for the real-time data.

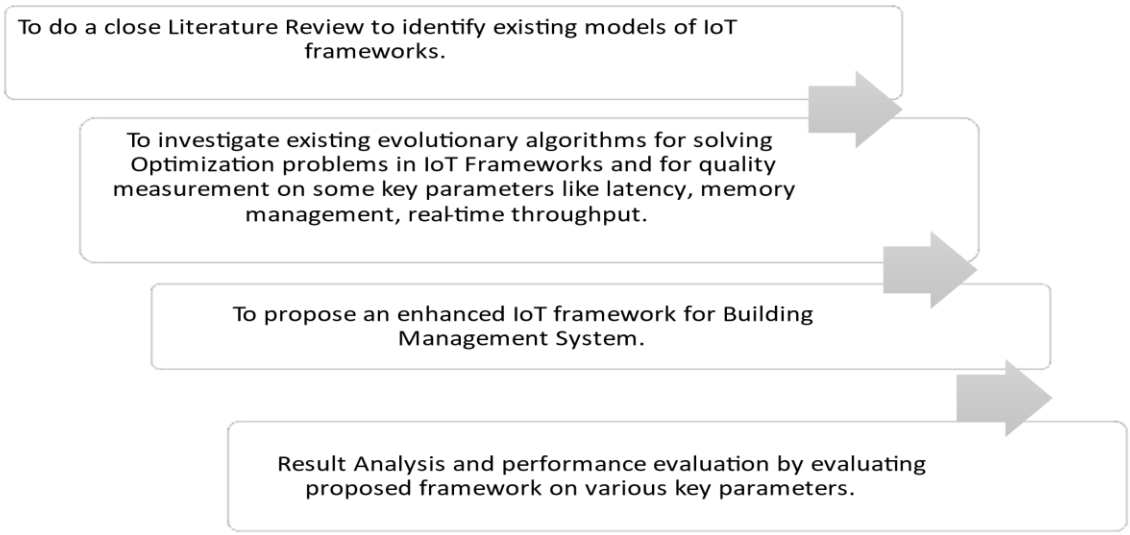
To do a close Literature Review to identify existing models of IoT frameworks.
To investigate existing evolutionary algorithms for solving Optimization problems in IoT Frameworks and for quality measurement on some key parameters like latency, memory management, real-time throughput.
To propose an enhanced IoT framework for Building Management System.
Result Analysis and performance evaluation by evaluating proposed framework on various key parameters.

Figure 2. Comprehensive Image of Methodology (Source: Self)

6 EXPECTED OUTCOME

The expected outcome should provide the trust that the QoS information will be kept secure and the collection and monitor of data would be secure for both entities and processes. Basically, two different types of smart contracts are given to guarantee the trust in QoS properties measurement which are:

1) Allowing the various operations of OS in QoS measurements.
2) Managing IoT services to be acquired in respective BMS.

The solution is common and capable of being scaled for any IoT services and IoT device in Building Management Systems and its applications.

REFERENCES

Abdallah, M., Jaber, T., Alabwaini, N. and Abd Alnabi, A., 2019, April. A Proposed Quality Model for the Internet of Things Systems. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 23-27, IEEE.

Aldrich, J., Chambers, C., Sirer, E.G. and Eggers, S., 1999, September. Static analyses for eliminating unnecessary synchronization from Java programs. In *International Static Analysis Symposium*, pp. 19-38, Springer, Berlin, Heidelberg.

Ammar, M., Russello, G. and Crispo, B., 2018. Internet of Things: A survey on the security of IoT frameworks. *Journal of Information Security and Applications*, *38*, pp.8-27.

Asghari, P., Rahmani, A.M. and Javadi, H.H.S., 2020. Privacy-aware cloud service composition based on QoS optimization in Internet of Things. *Journal of Ambient Intelligence and Humanized Computing*, pp.1-26.

Bogda, J. and Hölzle, U., 1999, October. Removing unnecessary synchronization in Java. In *Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pp. 35-46.

Baccelli, E., Gunes, M. and Schmidt, T.C., 2008. Operating Systems for the IoT–Goals, Challenges, and Solutions. In *Proc. IPSN*.

Cuka, M., Elmazi, D., Obukata, R., Ozera, K., Oda, T. and Barolli, L., 2017, March. An integrated intelligent system for IoT device selection and placement in opportunistic networks using fuzzy logic and genetic algorithm. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 201-207, IEEE.

Cheruvu, S., Kumar, A., Smith, N. and Wheeler, D.M., 2020. IoT frameworks and complexity. *Demystifying Internet of Things Security*, pp.23-148.

Djaker, A.B., Kechar, B., Ibn-Khedher, H., Moungla, H. and Afifi, H., 2020, June. Scalable and Cost Efficient Maximum Concurrent Flow over IoT using Reinforcement Learning. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 539-544), IEEE.

Goudarzi, M., Wu, H., Palaniswami, M. and Buyya, R., 2020. An application placement technique for concurrent IoT applications in edge and fog computing environments. *IEEE Transactions on Mobile Computing*, *20*(4), pp.1298-1311.

Hassan, W.H., 2019. Current research on Internet of Things (IoT) security: A survey. *Computer networks*, *148*, pp.283-294.

Iwendi, C., Maddikunta, P.K.R., Gadekallu, T.R., Lakshmanna, K., Bashir, A.K. and Piran, M.J., 2020. A metaheuristic optimization approach for energy efficiency in the IoT networks. *Software: Practice and Experience*.

Izci, D., Ekinci, S., Eker, E. and Kayri, M., 2020, June. Improved manta ray foraging optimization using opposition-based learning for optimization problems. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-6, IEEE.

Javed, F., Afzal, M.K., Sharif, M. and Kim, B.S., 2018. Internet of Things (IoT) operating systems support, networking technologies, applications, and challenges: A comparative review. *IEEE Communications Surveys & Tutorials*, *20*(3), pp.2062-2100.

Kawachiya, K., Koseki, A. and Onodera, T., 2002. Lock reservation: Java locks can mostly do without atomic operations. *ACM SIGPLAN Notices*, *37*(11), pp.130-141.

Kiruthika, J. and Khaddaj, S., 2015, August. Software quality issues and challenges of internet of things. In *2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pp. 176-179, IEEE.

Li, W., 2019. Application of Economical Building Management System for Singapore Commercial Building. *IEEE Transactions on Industrial Electronics*, *67*(5), pp.4235-4243.

Lu, Y. and Cecil, J., 2016. An Internet of Things (IoT)-based collaborative framework for advanced manufacturing. *The International Journal of Advanced Manufacturing Technology*, *84*(5-8), pp.1141-1152.

Minoli, D., Sohraby, K. and Occhiogrosso, B., 2017. IoT considerations, requirements, and architectures for smart buildings—Energy optimization and next-generation building management systems. *IEEE Internet of Things Journal*, *4*(1), pp.269-283.

Oti, A.H., Kurul, E., Cheung, F. and Tah, J.H.M., 2016. A framework for the utilization of Building Management System data in building information models for building design and operation. *Automation in Construction*, *72*, pp.195-210.

Patel, K.K. and Patel, S.M., 2016. Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, *6*(5).

Ren, X., Zhang, Z., Chen, S. and Abnoosian, K., 2021. An energy- aware method for task allocation in the Internet of things using a hybrid optimization algorithm. *Concurrency and Computation: Practice and Experience*, *33*(6), p.e5967.

Sangaiah, A.K., Hosseinabadi, A.A.R., Shareh, M.B., Bozorgi Rad, S.Y., Zolfagharian, A. and Chilamkurti, N., 2020. IoT resource allocation and optimization based on heuristic algorithm. *Sensors*, *20*(2), p.539.

Singh, M. and Baranwal, G., 2018, February. Quality of service (qos) in internet of things. In *2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, pp. 1-6, IEEE.

Sivaraman, V., Gharakheili, H.H., Vishwanath, A., Boreli, R. and Mehani, O., 2015, October. Network-level security and privacy control for smart-home IoT devices. In *2015 IEEE 11th International conference on wireless and mobile computing, networking and communications (WiMob)*, pp. 163-167, IEEE.

Srinidhi, N.N., Kumar, S.D. and Venugopal, K.R., 2019. Network optimizations in the Internet of Things: A review. *Engineering Science and Technology, an International Journal*, *22*(1), pp.1-21.

Suryanegara, M., Prasetyo, D.A., Andriyanto, F. and Hayati, N., 2019. A 5-Step framework for measuring the Quality of Experience (QoE) of Internet of Things (IoT) services. *IEEE Access*, *7*, pp.175779-175792.

Tian, Z., Luo, C., Qiu, J., Du, X. and Guizani, M., 2019. A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics*, *16*(3), pp.1963-1971.

Weber, R.H., 2009. Internet of things–Need for a new legal environment?, *Computer law & security review*, *25*(6), pp.522-527.

White, G., Nallur, V. and Clarke, S., 2017. Quality of service approaches in IoT: A systematic mapping. *Journal of Systems and Software*, *132*, pp.186-203.

Wijayasekara, D., Linda, O., Manic, M. and Rieger, C., 2014. Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions. *IEEE Transactions on Industrial Informatics*, *10*(3), pp.1829-1840.

Viriyasitavat, W., Da Xu, L., Bi, Z., Hoonsopon, D. and Charoenruk, N., 2019. Managing qos of internet-of-things services using blockchain. *IEEE Transactions on Computational Social Systems*, *6*(6), pp.1357-1368.

Zhang, Y., Shao, S., Ji, M., Qiu, J., Tian, Z., Du, X. and Guizani, M., 2020. An Automated Refactoring Approach to Improve IoT Software Quality. *Applied Sciences*, *10*(1), p.413.

Zhang, Y., Shao, S., Liu, H., Qiu, J., Zhang, D. and Zhang, G., 2019. Refactoring java programs for customizable locks based on bytecode transformation. *IEEE Access*, *7*, pp.66292-66303.

Zhang, Y., Dong, S., Zhang, X., Liu, H. and Zhang, D., 2019. Automated refactoring for stampedlock. *IEEE Access*, *7*, pp.104900-104911.

Zhao, W., Zhang, Z. and Wang, L., 2020. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Engineering Applications of Artificial Intelligence*, *87*, p.103300.